Mobile Development

IOS Core Data

Admin

- Quiz
- Current Events

So you want to save your data

- Suppose you are writing an iOS app and you want to save your data
  - What are your options?

So you want to save your data

- Suppose you are writing an iOS app and you want to save your data
    - What are your options?
        - Save offsite
            - In the cloud
            - Using web services
        - Save locally
            - Using text files
            - Using Archiving
            - Using iOS core data

Text files

- You know the good and bad of text files
    - Like?

Archiving

- Use NSKeyedArchiver
  - Provides serialization of objects.
  - Write out objects with their state
  - Much like java serializable interface.

Core Data

- Core Data
  - Purpose: use structured databases without knowing SQL

- Uses SQLite
  - Lightweight sql server
    - Library + structured file

iOS Core Data

- Role of core data:
    - Provide the mapping between table and objects
    - Objects
        - Each row in table is an instance
        - Each column is an instance variable/property

Creating a model

- Walk through creating a model.
    - Note limited data types supported
    - Most of it is pretty straight forward right?

All the other data

- Everything that is not in the list of data types
    - Needs to be transformable
    - Saved as NSData object.
    - Transformables need a transformer
        - Not 'more than meets the eye'
        - Transformer extends NSValueTransformer
            - Needs three methods
            - See page 434 of chap 23.

How do we hook it up?

- How do we hook up the transformer?
    - Data model inspector
    - Walk through it.

Core Data Entities

- Each entity in a Core Data model
  - Akin to a table.
  - Relationships between entities
    - Represented as pointers.
    - One to one
      - A pointer from object1 to object2
    - One to many
      - A pointer from object1 to an NSSet of instances
    -

Setting up a relationship

- Walk through setting up a relationship

NSManaged Object

- ## When data comes out of Core Data

  - It comes as an NSManaged object

  - Subclass of NSObject

    - Knows how to play nice with rest of Core Data

  - Might need to subclass NSManaged object

    - If you need more than a data container

    - If adding this to existing project – BNR has odd suggestion

    - Walk through creating data subclass

      - Select entity – new file – NSManagedObject subclass

NSFetchRequest

- This is how we will get the data
  - See example project

Reading

- Read chapter 23 in ios 4th editions
  - Chapter 16 in 3rd edition has most of this.